DOCUMENT RESUME

ED 145 728 FL 009 010

AUTHOR Wright, James R.
TITLE Computer-Based Linguistic Analysis.
PUB DATE Oct 77
NOTE 21p.; Paper presented at the Mountain Interstate Foreign Language Conference (October 1977); Not available in hard copy due to marginal legibility of original document

EDRS PRICE MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS *Componential Analysis; *Computational Linguistics; *Computer Programs; Computers; Discourse Analysis; English; Generative Grammar; Language Research; Linguistic Theory; Morphology (Languages); Phonology; Semantics; *Structural Analysis; Syntax; *Transformation Generative Grammar

ABSTRACT

Noam Chomsky's transformational-generative grammar model may effectively be translated into an equivalent computer model. Phrase-structure rules and transformations are tested as to their validity and ordering by the computer via the process of random lexical substitution. Errors appearing in the grammar are detected and rectified, and formal correctness of the grammar is verified. Non-trivial semantic problems arise when predictable compatible lexical items fail to interchange, and interesting syntactic problems occur when various related transformations interact. Applications of the computer to linguistic analysis (with actual computer output information) are shown on the following levels: (1) phonological: computer-generated phonotactically correct language simulations, including English, Spanish, Turkish, Cherokee, Japanese, and Chiquitlan Mazatex; (2) morphological: derivational prefixation and affixation of contemporary American English, with focus upon formalizing the complementary distribution factors of allomorphs; (3) syntactic: generation of English kernel sentences, a semantic feature lexicon, and the application and results of twenty representative transformations; (4) semantic: investigation of the degree of lexical interchangability tolerated within the boundaries of the eight English parts of speech. Non-computer-users are shown the ease with which neophytes can enter into this area of linguistic research and the results which they can expect to obtain immediately.
(Author/CFM)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

James L Wright

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED145728

Computer-based Linguistic Analysis

At present, digital computers are proving to be valuable research tools for linguists engaged in the investigation of the deep structures of natural languages. Although not an essential tool for either the construction or the testing of formal grammars, the computer can nevertheless greatly reduce the amount of tedium involved in such a project, particularly in the area of producing large volumes of output data for subsequent human evaluation and revision. A desirable side effect often occurs when the output suggests novel areas for further investigation which might never have otherwise occurred to the linguistic analyst.

This paper summarizes some of the projects undertaken during the past academic year, and touches upon computer-assisted linguistic analysis on the following language levels; phonological, morphological, syntactic and semantic. Everything which follows in this paper is based upon one underlying principle; a language consists of a systematic set of patterns and well-defined sets of lexical items which may be drawn from in order to realize, or fill, these slots. As a simple example, imagine five envelopes, labelled as follows; two DETerminers, two NOUNs, and one VERB. Each envelope contains a large number of slips of paper upon each of which a single lexical item is printed. The envelopes are now arranged in a horizontal row in the following left-to-right linear order; DET NOUN VERB DET NOUN. A person (or machine) extracts a single item from each of the five envelopes, and places the selected item on top of the envelope from which it came. It then follows that if the pattern represented by the arrangement of the envelopes is acceptable in terms of the linguistics system which it represents, and if the members within each set have been assigned valid labels, the sentence string which results will be gramatically correct, although its meaning may be aberrant for a number of reasons. If the string is not gramatically correct, an

FL009010

BEST COPY AVAILABLE

investigation is made to determine whether it is the pattern itself or a set member which is at fault. The error is corrected and the process repeated until satisfactory results for the experiment are obtained.

The programming language in use in the projects described herein is SNOBOL3. This language was selected for two basic reasons. In the first place, SNOBOL is a string-manipulating language by nature. Thus, it is ideally suited for dealing with pattern manipulations or transformations in a manner compatible with and parallel to the treatment employed by post-structural linguists. Secondly, the language was locally available. One of the staff consultants of the computer center on campus was familiar with both SNOBOL and with the basic concepts of linguistic theory and analysis, as well as with the rudiments of transformational-generative grammar. At the outset, it was only necessary to describe to him the type of data to be manipulated, the operations desired and the target results. The consultant immediately designed a basic SNOBOL program for achieving these goals.

Let us first observe an application of this program at the phonological level. On this basic language level, two types of language simulations were synthetically developed. One type involved listing as many as possible of the actually-existing syllables of a given natural language and then combining these at random into words of the same number of syllables as those actually occurring in the language. The results then would sound like words in the real language, but would be devoid of any semantic content except that which was coincidental.

Syllable-Generated Simulations

Turkish: čičekčiler šekerjilerden čičekler tašidi
'The florists carried flowers from the confectioners.'

Turkish: dičaler ǰimšiver gune alküzleg ǰuslerdamlen
Simulation: kuzǰuk sek yoror elzeli izdem čift kuzal lenlarček

Japanese: anata we nara ga hanasi o kiita to sinzita
'You believed that Nara heard the story.'

Japanese Simulation: rowaga i dimori fiyua itiuko hidai nipu aio isuu iyoi aamana tozoda sobii tuwao poare iwo beru kanat tokorawa

Hawaiian: tahuwai la a tahuai wai la enu nene la a pili koo lua la pututui lu a ite toe la hanu lipo ita paalai au we ta hua la
(Popular song written by Leleiohaku, 1936)

Hawaiian Simulation: ama nike alina ma nanaa tahe timeate halali lehomoke tuka mopoepou kopipona kukakunco liiti momatiki ane wumeunepa

Mandarin Chinese: ta zeleifan yikwian wo ǰiu zou le
'I left before he ate.'

Chinese Simulation: nunpaočaya lin tse ssu kenghsueh tiao to tan min yuanloo niaoniu pan tog ǰunhsiang čiensančia

Cherokee: tsalagi sikwoya duninodi usinuliya adasu
'Cherokee Sequoya October instantly war-club'

Cherokee Simulation: tsoliyia tluhduhwetletsu nuhnahasa tsekwogi kwitseo hunetlitsu segate hadlagu kwusite tasetluhna mikwuhe

Chiquinuitlan Mazatec: cakace nači hčinka š?i ?nti
'The little man bought the old horse.'

Mazatec Simulation: sqa tlantlan sequinusesqa sakenkihta hiatline huicuqa čonkaka etenta nuquessqa čovatue nyankuqmi čato kinate

As expected, the above simulations do capture the flavors of the languages which they mirror phonetically. They could not fail to do so, as actually-occurent items are employed. Though the results are phonologically semi-predictable and semantically void, they serve to actively stimulate the imagination of the linguist confronted with the task of making up small practice or examination problems in linguistic analysis.

In contrast, the samples which follow were generated on an even

lower level. This time, individual phonemes were combined to form the syllables, which above were preset, and then the created syllables were combined to form 'words'. Strict phonotactic patterns of the language being simulated were maintained throughout.. For example, to perform a simulation of English at this level, complete charts were compiled of all existent syllable-initial and syllable-final consonant clusters. These clusters were controlled so that they could only occur randomly in slots appropriate to real English. Current canonical syll ble shapes were exclusively employed. Predictably, if the phonotactic restrictions imposed were correct for actual English and maintained, the results should be readily pronouncable by native speakers of English. These could form a ready reserve lexicon of potential English words, which could be drawn from whenever a novel word was required in the language. One deficiency existing in contemporary American society is the area of singular non-sexist pronoun forms. What could be more neutral in selecting or composing these than a non-chauvinist computer?

Phone-Generated Simulations

English e joa varp sǒjglinned er proiǰuti eived aisedbims.
Simulation: ers bloent katsploes rollsploaved harfedka wat tu randarls.
ta edlas upuked, thilled, dumugged, ex atslo neazzed sib starps. if heerms dansponged, neld thok wabsmoots.

Japanese otama damogi uholu kuyezi a nua esua ita edu vuwayu unelo
Simulation: masaro ubegi narinhau rihayu ǰinika fayasi yokayaku katabaki

'Ganakaruku': sumufufi nokuuki ačaga faua riibi sounimo titilo dunuri
zičučora yohifulo agapuki iuoi kiyiu bubapu uwupa haduba

In generating the above strings it was necessary to represent certain IPA symbols, not available on an IBM keypunch, by arbitrary available symbols. Numbers and various punctuation marks and mathematical

operators are available, and this poses no real problem other than that of arbitrarily deciding what is to represent what, and then keeping the equivalencies straight. One practical value of the phone-generated simulations lies in the verification of phonotactic rules of real natural languages such as the rules governing the members and sequences of consonant clusters.

On the morphological level, the derivational word-making processes of English and Spanish were explored. The computer input lexicons on this level consisted of sets of free and bound bases, and inflectional and derivational affixes. One of the goals of this experiment was to attempt to measure the degree of interchangeability of morphemic alternants, i.e., allomorphs of the same morpheme, by discovering factors of complementary distribution which have thus far been eluding linguists. On this level, a large corpus is of tremendous advantage when attempting to characterize an entire language system. Unlike the words in the sound-simulated language reviewed above, morph-generated words and word strings are relatively intelligible. The degree of intelligibility depends upon the tensing or relaxing of semantic restrictions as well as the degree of transparency of the word roots employed, i.e., whether or not Greek, Latin, Anglo-Saxon, etc., roots are used or only those more readily recognizable by morphologically-naive native speakers. Although not thus far attempted, an experiment could be set up to test various speakers' abilities to decode novel morph-generated words by decomposing them and analyzing their constituent morphs. It was found, not surprisingly, that randomly-affixed prefixes forming novel words were more easily processed and decoded by readers than were words formed by derivational suffixation. This is due in part to the linear nature of language and partly to the fact that with derivational affixation frequently

a change in grammatical part of speech is involved.

Morph-Generated English

That hypercumb multithighfat quasicynic retrorubbed the transweird offdwarf. An orphanswine shesqualor rebit orand blastobliterated the thinthin codelightburger protofoulwards. This cushfella astutened that halfdirty monorednecked swarthyster rip-offescently. An excesso-stutterer circumsmacked a quadristrangeimbicileer youngcrazywise.

Morph-Generated Spanish

Aquel transtonto sobrejoven olvidó a ese intrapodrido seminombre. Un trifiaroizo superestupidario insultará al malverdicionote retroorgullosamente. Escupircito respetaración sobredentro aquelese brutancia e insimpaticificaruelo al insultaraderonotecillo brutillamente. Madrecillotonazuelote miito, que requetefeo ancianaraderonacno: Ese machecezuelo tipecitillotezerón ataca al estupidecicachezueluco salvajazachonazecetemente debajomientrasadentro suavecitando frescamentezón.

In progress at the present time is a project involving the testing of semantically similar morphs in an attempt to group them together as allomorphs of a single morpheme. Do two morphs which have the same (approximate) semantic reading, which are both noun-to-adjective derivational suffixes, and which pattern similarly syntactically constitute a single morpheme? How can this be objectively measured and determined? If there indeed is evidence that they are allomorphs of the same morpheme, are they in complementary distribution or free variation? As an example of this type of problem, consider the derivational suffixes in the three words manly, foolish and childlike, each of which converts noun roots to adjectives. All three seem to share one crude semantic reading in common: 'like a, characteristic of a.' By having a computer construct and furnish a variety of examples, much time is saved which can then be profitably spent in analyzing the outputted forms.

	-ish	-ly	-like
1.	mannish	manly	manlike
2.	?fatherish	fatherly	fatherlike
3.	childish	#childly	childlike
4.	waspish	*wasply	wasplike
5.	apish	?apely	apelike
6.	devilish	?devilly	devillike
7.	*shapish	shapely	?shapelike
8.	*hourish	hourly	?hourlike

* = not acceptable in author's ideolect
? = not clearly acceptable, but not bad, either
\# = rare, but occurrent, though not in author's ideolect

The -ly suffix in items 7 and 8 is clearly not the same -ly on the first six items in the second column, and thus should be discarded as a potential allomorph of the morpheme mentioned above. There seems to be no discernible consistent phonological or semantic patterning here, though perhaps more time and insight may reveal a simple solution.

To date, three projects have been initiated on the syntactic level involving computer application. The first deals with English conjunctions. These seem to be rather consistently slighted by many traditional grammarians, who usually describe conjunctions as connectors or connective elements. Some grammars subdivide conjunctions according to the type of sentence components which they connect.

Coordinating Conj.:	That professor is hard, but fair.
Correlative Conj. :	He not only drank, but also stank.
Comparative Conj. :	Gino is as stupid as Zorba.
Subordinating Conj.:	Unless she improves, she will fail.

By utilizing the computer to furnish the examples, the claimed interchangeability of members of certain conjunction subclasses may be checked for accuracy and completeness of membership. As an example, the words since, while, when, and until are presumably interchangeable when functioning as conjunctions. By setting up one test frame sentence with an appropriate slot, an entire group may be tested simultaneously.

Test Frame: _____ we ate it we were sick.
[since, while, when, until]

1a.	?Since we ate it we were sick.	2a.	Since we ate it we have been sick.
1b.	While we ate it we were sick.	2b.	While we ate it we got sick.
1c.	When we ate it we were sick.	2c.	When we ate it we got sick.
1d.	Until we ate it we were sick.	2d.	?Until we ate it we got sick.
1e.	Since we ate it we got sick.	2e.	Because we ate it we got sick.

The following items involve time periods: 1b,1c,1d,2a,2b,2c,2d.

1a seems to be grammatical if since means 'for what reason'.

2a is ambiguous, as since may mean 'cause for result' or 'from then on'.

2d is grammatical if until means 'up to the time that'.

Hence we note that complete interchangeability for since, while, when, and until, although claimed in some traditional grammars, implied in others, is simply not the case. Similar problems arise when the test slot is moved.

Test Frame: We ate it _____ we were sick.
[since,while,when,until,(because)]

Test Frame: One guy coughed _____ she sang.
[as, after, before, during, when, while]

3a.	One guy coughed as she sang.	4a.	*One guy coughed as her singing.
3b.	One guy coughed after she sang.	4b.	One guy coughed after " " " ".
3c.	One guy coughed before she sang.	4c.	One guy coughed before her " ".
3d.	*One guy coughed during she sang.	4d.	One guy coughed during " " " ".
3e.	One guy coughed when she sang.	4e.	*One guy coughed when her singing
3f.	One guy coughed while she sang.	4f.	*One guy coughed while her " " .

Clearly, here, as above, the test items are not interchangeable members of a single set. By utilizing this approach as a point of departure, additional conjunctions can be added to the set being tested. The goal emerging from this experiment is the eventual redefinition of traditionally classified conjunctions, the experiment clearly indicating the necessity of employing the intersection of form, function, and semantic content.

Students currently engaged in performing clinical speech therapy are using the computer to compile sets of articulation drills. Their main difficulty lies in getting the target sounds into short natural-sounding sentences containing knowledge compatible with that of the age groups of their clients. This involves both lexical inventory and pattern mastery. After determining which sound types are problematic and in which phonological contexts i.e., syllable/word onset, nucleus, or coda, word lists are compiled containing the target sounds in their target positions of occurrence. The words are next lexicalized according to part-of-speech categories, rechecked for their suitability for the involved clients, and then randomly inserted into kernel sentence patterns by the computer. The resultant potential drill sentences are then edited by the clinicians, but the computer can be programmed to either jumble the patterns and/or target items, or to produce them serially (or randomly) in homogenous groups.

Drill /k/$_1$. Word-Final /k/, Noun Slot(s)

1. This is a park.
2. This is (a) bark.
3. This is dark.
4. This is a dark horse.
5. This is a shark.
6. This is a mark.
7. This is a lark.
8. This is a narc.

It should be noted here that a valuable aid in constructing such drills, and one which is simply not available at the present time, would be an Inverse English Dictionary (i.e., inverse dictionary of the Eng. language). In such a dictionary, all words are alphabetized according to their <u>last</u> letter, all words having the same last letter then being in turn alphabetized according to the penultimate letter. Hence, in

such a dictionary for English, the first page would resemble the following: a, b, c, d, ..., z, aa, ba, ca, ...za, aaa, ... zzz, etc. Such dictionaries exist for many languages other than English, and are of great value for finding and investigating word-final and word-medial letters, sounds, and sound sequences. In a language such as Spanish, in which verbs are listed with infinitival endings (-ar, -er, -ir) , this type of dictionary contains all Class I, Class II, and Class III verbs (with a few non-verbs interspersed) in the same area. Thus, the investigator can proceed directly to the area wanted, rather than to have to scan an entire conventional alphabetical dictionary and pick out items individually, and find everything he needs in one location. As one of the simplest of tasks for a computer to perform is the serialization of lists, it is fervently hoped that an inverse dictionary for English is soon to be forthcoming.

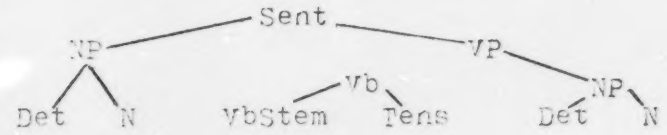
SNOBOL, being a string-manipulating language, is very well suited for constructing a computer model of transformational-generative grammar. Such a model is of great value to a linguist in performing his role as a grammar writer/tester/evaluator. The following section describes how Noam Chomsky's model has been translated into a computer equivalent. The computer outputs transformational-generative grammars which are both intelligible and acceptable, the latter in terms of both descriptive and explanatory adequacy.

The first component of a transformational-generative grammar is a set of Phrase-Structure Rules which delineate the Kernel Sentence Patterns of a language. Although not a formal part of a grammar, this set of rules is equivalent to, and frequently accompanied by Branching Tree Diagram(s) which show hierarchical layering of

constituent structure.

Sample Trans-Grammar Phrase-Structure Component

```
        #Sent#
1.  Sent  -->  NP + VP
2.  VP    -->  Vb + NP
3.  NP    -->  Det + N
4.  Vb    -->  VbStem + Tens
```



It is necessary to modify the above structural description in order for the computer to prosess and manipulate the information which it represents. This is accomplished by compressing the information into a single linear statement by employing paired sets of nested parentheses. Here, the latter indicate constituent relationships rather than options, as in Chomsky's model.

Computer-Equivalent Phrase-Structure Component

```
(  NP  (  DET  N   )  VP  (  VB  (  VBSTEM  TENS  )  NP  (  DET  N  )  )  )
1      5           5      2      3                3      4           4  2  1
```

Note: The numbers below the line are not used, but are included here to indicate relationships and identities of matched pairs of left and right parentheses; the lower the number, the higher the constituent in the structure)

The second component in Chomsky's formal transformational-generative grammar is the Lexicon, which specifies the grammatical set names and indicates items contained within these sets, i.e., morphemes and words.

Sample Trans-Grammar Lexicon, Early Style

Noun boy, girl, cat, dog, table, chair, tree, house, window, ...

VbStem hit, kill, stamp, buy, open, imitate, lose, caress, ...

(... indicates an open set, which may be extended indefinitely)

Sample Trans-Grammar Lexical Entries, Semantic Feature Style

dude --> [+Noun, +Singular, +Animate, +Human, ±Female, +Concrete, +Count, -Proper]

Big Mac --> [+Noun, +Count, +Proper, +Singular, -Animate, -Human, +Concrete]

The latter version of the lexicon was the one chosen for the computer model, as it enables the programmer to control semantic as well as syntactic compatibility. As in the case of the Phrase-Structure Rules, computer Lexical rules must be in strict linear form. A sample entry in the computer version's lexicon would look as follows.

DUDE N COUNT+ PROPER- SING+ ANIM+ HUMAN+ CONCRETE+ FEM+ FEM-
(Each entry in the lexicon is contained on a single IBM Card)
BIG MAC N COUNT+ PROPER+ SING+ ANIM- HUMAN- CONCRETE+

Component number three in both the versions of transformational-generative grammar is the set of Transformational Rules, which convert the basic structural patterns delineated by the Phrase-Structure Rules into variations of these, such as, for example, Negatives, Yes/No Questions, Passives, Clefts, etc.

Sample Transformational Rule, Syntactic Structures Version

T. PASSIVE (optional) NP1 + VbStem + Tens + NP2 ==> NP2 + BE + Tens + VcStem + PPS + by + NP1

This rule, when applied to a Kernel sentence of the type SVO, would have the following effect;

That kid + wrote + the obscene letter ===>
NP1 + VbStem/Tens + NP2

The obscene letter + was + written + by + that kid
NP2 + BE/Tens + VbStem/PPS + by + NP1

Passive Transformation, Aspects Version

SD: that kid wrote the obscene letter
NP VbStem/Tens NP
1 2 3 4

SC: 4 be 3 2 PPS by 1

(i.e., The obscene letter was written by that kid.)

The latter rule makes it instantly apparent that this rather complex transformation reverses the original order of the four constituents of the kernel sentence and also adds three new items (be, PPS, by). As might be anticipated, this will require several operations for the computer to manipulate the four and add the three in their correct respective slots.

Computer-Equivalent of Passive Transformation

```
SENTENCE *STRING1 " NP *STRING2 " NP " *STRING3*
STRING2 *(NPHRASE1)* *STRING4* VB ( "*STRING5* " V " *(STRING6*)=
STRING3 *(NPHRASE2)*=
SENTENCE = STRING1 " NP " NPHRASE2 STRING4 " AUX ( BE TENS ) "
SENTENCE = SENTENCE STRING5 " VB " STRING6
SENTENCE = SENTENCE " PPS BY ) ( NP " NPHRASE1 STRING# " ) "
```

Arbitrarily, the following output format was chosen, in order to illustrate with what was felt to be maximum clarity for neophytes to transformational-generative grammar the effects of transformations upon the structures to which they apply:

```
INPUT:  KS1  That student hates computers.
T-NAME: PASSIVE TRANSFORMATION
T-RULE: NP1 + VBStem+ Tens + NP2  ==> NP2+BE+Tens+VbStem+PPS+by+NP1
RESULT: Computers are hated by that student.
```

The actual formatted computer output is identical to the above, except that the Input Kernel Sentence information contains both the Structural Kernel Sentence Pattern (which each time is randomly chosen from the pattern inventory data section) and the lexical items which have randomly been inserted via computer from the available Lexicon data;

Formatted Computer Output, Passive Transformation

INPUT: KS1 (NP (DET (THAT) N (STUDENT)) VP (VB (V (VBSTEM (HATE) (TENS (S)) NP (N (COMPUTER) PL (S))))

T-NAME: PASSIVE TRANSFORMATION

T-RULE: NP1+VbSTEM+Tens+NP2 ==> NP2+BE+Tens+VbSTEM+PPS+by+NP1

OUTPUT: Computers are hated by that student.

As noted above in the discussion of the Computer Version of the Lexical Component, it was decided to exercise control of the intelligibility of the output sentences by building into the lexicon the semantic features requisite for matching items in sentences lexically to mirror what is possible in the real world. This control is variable from total restriction to total arbitrariness. The programmer decides upon the "Semantic Feature Minimum Acceptance Level" desired on a particular run of the program. Level 0 specifies that all items must be compatible within a sentence with regard to their semantic feature matched content, AS MUCH AS POSSIBLE. If no item is available which exactly matches, only then is the lexicon searched for total feature compatibility minus one. If found, the near-matching item is pulled, otherwise scanning continues to find an item containing all but two of the matching semantic features. To illustrate;

Semantic Feature Minimum Acceptance Level; Illustration

Level 0: That man married the pretty woman. (Total Lexical Restriction)

Level 2: That man married the charcoal-broiled woman.

Level 4: That man married the multilingual ape.

Level 6: That man married a boiling iceberg.

Level 8: That ruthless omlet married the obese hatred.(No Lex. Restricti

Thus far, the computer has proven of great value in verifying the accuracy and completeness of lexical entries, and in checking semantic correlations. It has played an equally important rule with regard to Phrase-Structure Rules and Transformational Rules, providing specific data for analysis as to formal completeness, accuracy, and correct rule ordering. In addition, it furnishes an automatic symbol count, for those who wish to utilize this evaluation metric for comparing different grammars.

On the semantic level, the computer has been utilized to artificially stimulate the imagination of would-be creative writers. Expanded expressiveness is stimulated when human-restricted views of the order apparent in the universe are abandoned for random combinations of meaning. A piece of good existing prose or poetry is analyzed and broken down into its syntactic patterns, or else a new piece is created from scratch. Lexical items are randomly substituted into the patterns. The lexicon can be controlled so that only synonyms or near-synonyms are interchanged, or ANY lexical item of a given word-class set may be inserted. Here are a few representative samples:

Original Poem

Strange foreheads ignor nonor and slow fate
As your spotted presence climbs the moaning showers.
Bows of smoke inhabit ancient tempers
And noble clods reduce the officious stairway.

Amble into father's panting tuba;
Breakfast over rusty porcelain visions.
In a tornado, the owls decide the message
While sleek conclusions forge uneasy dropouts.

Proverbs

All work and no play makes Jack a dull boy. (Original)
All booze and no action make Billie Carter a crazed boob.
All beer and zilch for chicks make the Grand Dragon a pathetic ass.
All lip service and nil compassion make King Lear a dour bore.

You can't make a silk purse out of a sow's ear. (Orig.)
You can't construct an epic poem out of an anteater's armpits.
You can't extract a chocolate cake from a politician's droppings.
You can't yank a martini out of an (un)civil servant's cleavage.

He who laughs last, laughs best. (Orig.)
The cat who gets stoned the most, giggles longest.
She who flips out incessantly, freaks out in vain.
He who belches for no reason, gets blown away like a fire hydrant.

A bird in the hand is worth two in the bush. (Orig.)
An ape in the back seat is not the same as nine in the basilica.
A buzzard in the bonnet is far better than dozens in the bowels.
A bra in the chops is better than scores in the closet.

Newspaper Headlines

Unbalanced Mexican kidnaps journalism prof

Shrieking Queen Elizabeth barbequed by frenzied gorilla

Rabid Cub Scout shoots slurping university chancellor

Lust-crazed gooo ole boy faked out by cretin frump

Popular Music Chord Chart (Random)

D7 Amin6 CMaj7 F#min9 E7#5#9 Ab7-5 Bmin7 GMaj6 Eb9+6 FMaj7 Bdim A#b5#9

Mexican Cantina

In a cantina
Of six tables
A prostitute strokes
The hand
Of a public accountant.
It is silent,
And the room is as empty
As a peasant's belly.
A galaxy of insects
Darts around the dirty lightbulb.
Jose cleans the ashtrays,
Flicking the dirt
Onto the floor.
That pair
With the faces of oxen
On a sweltering afternoon
Seems to float above his head
And evanesce into the mirror.
Finally, they leave
Without saying a word,
His hand brushing
Her right thigh,
Her thumb hooked
Into his belt loop.
The waiter closes the door,
Pours himself a drink,
And slouches at a table,
And his digits flex and unbend.
Against the ceiling,
A myriad of spectral images -
The coyote,
The alerted impala,
A butterfly,
A crocodile wolfing the ozone.
While the rooster silently practices
For the dawn not yet arrived.

- Manuel Labor

At the Cantina

Inside a Mexican cantina
Of gloomy depression
A chica caresses
The forearm
Of a medical student.
It is late
And the place is as void
As a politician's promise.
A squadron of flies
Zooms near the solitary light.
The waiter polishes the bar
Showering dust
Onto the stools.
The couple
With the expressions of pack mules
On a scorching midday
Appears to stretch beyond his shoulder
And fade into the shadows.
At last, they slip out
Without even nodding adiós,
His arm touching
Her undulating buttock,
Her hand buried
In his coat pocket.
Tío Pepe closes up,
Grabs a bottle,
And slumps down at the bar,
And his arms fold and open.
On the walls,
A forest of ghostly silhouettes -
The jaguar,
The sleepy doe,
The hawk,
An iguana snacking on the breeze -
As the cock stretches
For the day not quite here.

- Kay Pasa

As can be observed through a comparison of the two versions of the 'Cantina" poem, lexical substitution into well-defined patterns provides rich fields for semantic exploration. These two versions, which were not humanly edited after they were printed by the computer, illustrate the effect of careful control of lexical substitution. Members of the sets were semantically related, from near-synonyms to merely similar types of meanings. More relaxed restrictions result in itmes such as the 'proverbs', whereas complete unrestrictedness yields items such as the original poem on page 12, above, in which any overall meaning is dependent upon the experience of the reader, or is a sheer accident. By experimental substitution along the lines outlined above, would-be writers can actually visualize the various shades of difference which can result in expressiveness by manipulating and controlling various factors of a language system either independently or by observing various co-restrictions.

Computers cannot perform any operations which they are not instructed to do. They are incapable of making independent judgments. They can, however, be programmed to furnish output data along specific guidelines which can subsequently be evaluated, manipulated, edited, etc. The experimental applications outlined in this paper illustrate the suitability and adaptability of computers in general and the SNOBOL programming language in particular for manipulating language and performing various phases of linguistic analysis. The experience of this writer suggests that computer feedback is an invaluable aid for delving ever deeper into the infinite subtleties of human language systems.

REFERENCES

Chambers, John M., *SIGOL3 Manual*, Computer Sciences Dept., University of Wisconsin, Madison, Wisconsin, 53706.

Chomsky, Noam, *Syntactic Structures*, Mouton, The Hague, 1957.

Chomsky, Noam, *Aspects of the Theory of Syntax*, M.I.T. Press, Cambridge, Mass., 1965.

Doran, Robert W., *360 O.S. FORTRANIV Free-Field Input-Output Subroutine Package*, CS-79, Computer Science Dept., Stanford University, 1967.

Friedman, Joyce et al. *A Computer Model of Transformational Grammar*, Elsevier, New York, 1971.

Friedman, Joyce, ed. *Computer Experiments in Transformational Grammar*, CS-106, Computer Science Dept., Stanford University, 1968.

Friedman, Joyce et al. *Programmers' Manual for A Computer System for Transformational Grammar*, CS-115, Computer Science Dept., Stanford University, 1968.

Lieberman, D., ed. *Specification and Utilization of a Transformational Grammar*, IBM Watson Research Center, Yorktown Heights, N.Y., 1966.

Partee, Barbara H., P. Schachter, R. Stockwell et al. *Working Papers, U.C.L.A.-Air Force English Syntax Conference*, multilithed, 1967.

Ross, J.R., *Constraints on Variables in Syntax*, Ph.D. Dissertation, M.I.T., Cambridge, Mass., 1967.

Stockwell, R., P. Schachter, Barbara Partee et al. *Working Papers of the English Syntax Project*, U.C.L.A., mimeographed, 1967.

Zwicky, A.M., Joyce Friedman, Barbara Hall, and D. E. Walker, "The MITRE Syntactic Analysis Procedure for Transformational Grammars," *Proc. Fall Joint Computer Conference*, 1965, Vol. 27, Part 1, pp. 317-26.